

*Citation for published version:*

Jones, E, Alexander, J, Andreou, A, Irani, P & Subramanian, S 2010, GesText: Accelerometer-Based Gestural Text-Entry Systems. in *Proceedings of the 28th International Conference on Human Factors in Computing Systems (CHI'10)*. CHI '10, Association for Computing Machinery, pp. 2173-2182.  
<https://doi.org/10.1145/1753326.1753655>

*DOI:*

[10.1145/1753326.1753655](https://doi.org/10.1145/1753326.1753655)

*Publication date:*

2010

*Document Version*

Peer reviewed version

[Link to publication](#)

© ACM, 2010. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in CHI '10: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems April 2010 Pages 2173–2182 <https://doi.org/10.1145/1753326.1753655>

## University of Bath

### Alternative formats

If you require this document in an alternative format, please contact:  
[openaccess@bath.ac.uk](mailto:openaccess@bath.ac.uk)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# GesText: Accelerometer-Based Gestural Text-Entry Systems

Eleanor Jones<sup>1</sup>, Jason Alexander<sup>1</sup>, Andreas Andreou<sup>1</sup>, Pourang Irani<sup>2</sup> and Sriram Subramanian<sup>1</sup>

<sup>1</sup>University of Bristol,

Bristol, United Kingdom

{eleanor, jason, andreas, sriram}@cs.bris.ac.uk

<sup>2</sup>University of Manitoba,

Winnipeg, MB, Canada

irani@cs.manitoba.ca

## ABSTRACT

Accelerometers are common on many devices, including those required for text-entry. We investigate how to enter text with devices that are solely enabled with accelerometers. The challenge of text-entry with such devices can be overcome by the careful investigation of the human limitations in gestural movements with accelerometers. Preliminary studies provide insight into two potential text-entry designs that purely use accelerometers for gesture recognition. In two experiments, we evaluate the effectiveness of each of the text-entry designs. The first experiment involves novice users over a 45 minute period while the second investigates the possible performance increases over a four day period. Our results reveal that a matrix-based text-entry system with a small set of simple gestures is the most efficient (5.4wpm) and subjectively preferred by participants.

## Author Keywords

Gestural input, mid-air, text-entry, accelerometers.

## ACM Classification Keywords

H5.2 [Information interfaces and presentation]: User Interfaces - Graphical user interfaces.

## General Terms

Human Factors.

## INTRODUCTION

Embedded motion sensors are becoming increasingly common in many devices, ranging from generic consumer electronics to more specialized hardware dedicated to specific tasks. It is not surprising that 2007 was coined the year of the accelerometer [10]. With a five-fold drop in sensor costs [10], the ubiquitous accelerometers have allowed the general population to become familiar and comfortable with their use for a variety of tasks.

While devices with accelerometers are geared toward supporting primary tasks, they are also used for auxiliary functions such as entering text. For example, the Nintendo Wii, though primarily used as a game controller, is still required

to support text entry so that players can input their names, send messages to other players, or browse the internet.

Researchers have previously designed accelerometer based text-entry techniques, such as TiltText [20], and techniques for distant text entry on large displays [18]. However, these systems rely on auxiliary input mechanisms or devices to aid text-entry. For example, TiltText requires a keypad in conjunction with the accelerometer. Shoemaker et al [18] required an IR-enabled device with accelerometers for carrying out text-entry. While these systems target specific hardware platforms, an accelerometer based text-entry system should ideally be hardware independent. This would allow for a wider range of devices, such as tilt-enabled watches or remote controls to facilitate the text-entry task.

The ubiquity of text entry warrants an investigation of this task on accelerometer-only devices. Text entry is, fundamentally, a series of selections from a set of options. The findings of this research therefore go far beyond this task and can be applied to any accelerometer-based selections.

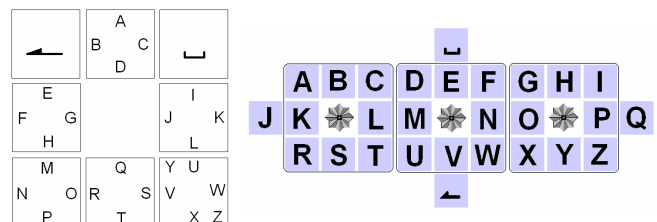


Figure 1: Proposed GesText systems, (left) matrix-based layout, (right) tri-center layout

This paper starts by analyzing the factors that influence the design of accelerometer-only text-entry systems. It then reports the results of two pilot studies that identify the extent to which designers can harness accelerometers within the limits of human wrist motion. These results assist with the design of GesText, a class of novel accelerometer-only text-entry systems. This accelerometer-only investigation ensures device independence and increased generalization of design recommendations by eliminating compounding design factors (e.g. the use of buttons and their size, placement etc). In addition this enables GesText systems to be distance independent (unlike IR-based systems) and potentially ‘eye’s free’. In two formal experiments it compares a matrix-based gesture layout and tri-center layout (see Figure 1) and shows that users can reach up to 5.4wpm when using a design that uses multiple simple gestures for character selection, as exemplified by the matrix-based layout.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.

Copyright 2010 ACM 978-1-60558-929-9/10/04...\$10.00.

The main contributions of this paper are: 1) a description of three classes of factors (human, accelerometer and interface) that can influence the design of accelerometer-based text-entry systems; 2) the limits of a set of distinct wrist-based gestures with accelerometers; 3) GesText: a class of novel text-entry designs that only rely on tilt input; 4) a short term and long term evaluation comparing two GesText designs, for both user efficiency and subjective satisfaction.

## RELATED WORK

### Accelerometer Based Input

Research into accelerometer based input can be classified into two categories: the utility of accelerometer-enabled devices in higher level applications, and low level studies that identify compelling properties of tilt for interaction.

Rekimoto's work was one of the earliest systems to propose accelerometer-based input as a means of controlling virtual objects on the screen [16]. Rekimoto demonstrated that designers can map angular tilt input in either a continuous or discrete manner. As a result, tilt values could be effectively used for carrying out actions such as selecting items from pull-down menus, interacting with scroll bars, panning or zooming around a spatial workspace, or even to perform more complex tasks such as 3D object manipulations.

Hinckley et al. [5] demonstrated a higher-level use of accelerometers by capturing tilt input for automatic screen orientation and scrolling applications, a feature that is now common on many digital consumer electronics. Weberg et al. [19] showed that with tilt input, users can control a cursor on a mobile device, by mapping the degree of tilt to the rate of cursor movement. Eslambolchilar et al. [2] showed that scrolling documents with Speed Dependent Automatic Zooming (SDAZ) [6] was effective when using tilt to trigger navigation operations. Other high-level applications for tilt include interacting with menus [13], controlling object placement on a digital tabletop [8], or entering text [14,20].

Researchers have also studied properties of tilt input at a more basic level, such as the effect of feedback on tilt input [12], the minimal achievable resolution with tilt [1], and various methods for discretizing and controlling tilt values [15]. For instance, Oakley and O'Modhrain [12] describe a tilt-based system with tactile augmentation for menu navigation and show that tactile feedback can be more effective than visual feedback, particularly when the device is turned away from the users' field of view. Crossan et al. [1] investigated the resolution of tilt input for controlling a cursor when the device is attached to the wrist, such as on a watch. They showed that error rates dropped significantly when the targets selected had a width of 9° or more. A systematic study by Rahman et al. [15] analyzed the design space of wrist-based interactions by studying separately the fluidity of control along the three axes of wrist movement: flexion/extension, pronation/supination, and ulnar/radial deviation. Their results reveal that users had the most control on the axis of pronation/supination and can achieve a resolu-

tion of target selection as low as 5°. While their work looked at each of the wrist axes in isolation, they did not investigate the properties of wrist movement along the combination of these axes or the resulting design space for the task of text-entry.

### Gesture Based Text-Entry Systems

Researchers have demonstrated the effectiveness of tilt input for text-entry systems [14, 20]. With TiltType [14] and TiltText [20], users enter a character by pressing a keypad button and then tilting the device in the right direction to enter the letter. Wigdor et al. [9] compared TiltText to conventional MultiTap and found that TiltText is around 23% faster than MultiTap (11.8wpm and 10.1wpm respectively).

Shoemaker et al. [18] investigated mid-air text-entry using tilt. Their study was motivated by the need to enter text on large wall based systems at a distance away from the medium. Shoemaker et al. [18] suggest that for a mid-air text-entry technique to be effective, distance independence (i.e. not requiring users to be in proximity to the display) is a key factor. Shoemaker et al's systems do not consider the possibility of entering text when only relying on accelerometers and achieved input rates between 6 and 10wpm.

Sazawal et al. [17] introduced Unigesture, a one-handed text-entry system for tilt-based devices. The design of Unigesture was also motivated by the need for small devices to perform text-entry, even in the absence of additional input buttons. The Unigesture layout divides an alphabetic list into seven zones. Users enter text by tilting to the appropriate zone, analogous to rolling a marble on a plate. Once a few letters are entered, the system infers the possible word. As a result, Unigesture is dependent on a dictionary. Users achieved a text-entry rate of 3.33wpm with Unigesture.

### Discrete Gesture Based Interactions

Tilt input is largely gesture based and this form of input has shown relative success for several types of tasks. For example, gesture-based menu interaction, such as marking menus, allow for item selection via pen strokes in the direction of a desired menu item [7]. The interaction becomes even more powerful as users grow more familiar with the strokes, thus relying less on any visual feedback or markers [21]. Selection is scale invariant so that a menu item is selected only based on the direction of the stroke. Studies have shown that expert users can perform menu selection 3.5 times faster with gestures, such as with marking menus, than when using regular pie menus [7]. Since marking menus largely depend on wrist movement, albeit in 2D, we inherit some of the design properties from this form of menu interaction for candidate designs of GesText systems.

## DESIGN FACTORS FOR ACCELEROMETER-ONLY TEXT-ENTRY SYSTEMS

This section develops a set of design factors to aid the development of accelerometer-based GesText systems. We describe these factors based on properties of the human operator, the hardware sensors and the interface.

## Human Considerations

**Scale of Gestures.** Accelerometers can detect gestures ranging from small scale movements, such as the deflection of a finger, to whole body motion. While large scale gestures enable greater scope of expression (e.g. tracing letters in air) small simple gestures are often easier to perform accurately and detect reliably. Gestures that require large or complex movements should be avoided as they increase the likelihood of physical and mental fatigue [11].

**Range and Axes of Motions.** Gestures are comprised of physical movements along single or multiple axes in three dimensional space. GesText depends to a large extent on wrist motions. These motions are limited by the range-of-motion (ROM) along three axes, which vary considerably from one axis to another [3]. A study on the dexterity of wrist motions with tilt input by Rahman et al [15] suggests that tilt input should rely more heavily on the axes that afford the highest level of ROM, namely the axes of pronation/supination and flexion/extension [3].

**Gesture Space.** Large scale gestures (e.g. drawing letters in mid-air) are inevitably performed in public spaces while small scale gestures (e.g. wrist rotations) are usually performed in personal or private space. Designers should consider the device's context of use, to ensure that the gesture set is socially and contextually acceptable.

**Learnability.** Prior studies have shown that a small set of simple gestures (such as vertical tilting and horizontal rolling) result in fewer errors as they are easier to memorize and reproduce [1, 15]. In systems that use unique gestures for each character item, there is a learning-recognition trade-off between gesture sets and optimal gesture recognition. Ideally, the simplest gestures afford a higher degree of learning [7, 21] and should be employed whenever possible.

## Accelerometer Considerations

The granularity of motion detectable by accelerometers means that shape-drawing input methods are potentially viable. However, since motion (degree of tilting along axes) is calculated from acceleration, sudden movements can cause changes in the tilt readings. Furthermore, a rapid gesture in one direction can generate a counteractive force in the opposite direction complicating interpretation (such as when abruptly stopping a high speed movement). Due to this difficulty in reliably differentiating velocity and range of motion, tilt gestures should be separated from other movements to improve detection. Relative positioning is also challenging and its convoluted calculation method means that it is prone to cumulative (drift) errors. Accurate relative positioning in 3D space can be achieved using additional sensors (for example, [18]). Absolute positioning requires calibration through an additional external input, such as an initial or continuous geospatial position.

**Reducing Gesture Recognition Ambiguities.** To ascertain whether a gesture has actually been performed as intended, movement along multiple axes can be combined to reduce any ambiguities. However, this process is still not straight-

forward as it requires statistical analysis and pattern matching techniques rather than instantaneous co-ordinate or vector analysis to recognize gestures. Some devices such as the Nintendo Wii are therefore coupled with secondary sensors (e.g. the IR sensor bar) to enable greater confidence in gesture interpretation.

**Immutable vs. Evolving Gesture Sets.** In a similar manner to digital tablets with handwriting recognition, gesture sets are immutable or evolve through system training. Complex gestures may be unreliably reproduced between different users or even by a single user over time [15]. This necessitates a trainable, evolutionary style of recognition system to achieve an acceptable error rate. Conversely, immutable gesture sets should therefore contain only intuitive, axial gestures. This makes gesture sets comprised of basic tilting and rotating movements highly suited to public, walk-up and use systems where no user training is required.

**Gesture Input and Interpretation.** It is difficult for accelerometer-only input methods to differentiate text input gestures from meaningless movements. Tablet-based handwriting recognition does not suffer from this issue as the contact of the pen with the surface indicates input. The explicit pen-up signal, along with micro-pauses between character entry and the physical pen location can be used to delimit the start and end of characters. These spacing and lifting techniques are not easily transferable to accelerometer based gestural input. In the absence of a selection mechanism (e.g. a button), accelerometer based systems must use explicit signals to delimit character segmentation and gesture recognition. These may take the form of a delimiting gesture [4], default position/orientation or an intentional pause.

**Mapping of Gestures to Input.** Gesture input mapping can be divided into two primary categories: direct mappings and selection techniques. Direct mappings are 1:1 encodings of a gesture to a character. Selection techniques involve the reuse of a limited set of gestures in order to navigate to the desired input character on a virtual keyboard. Most commonly, the characters on a virtual keyboard are arranged in a fixed layout; however, it is also possible to use moving target selection techniques.

## Interface Considerations

**Virtual Keyboards vs. Freeform Gestures.** Drawing standard alphabetic characters requires little learning effort from the user, but the natural variation makes recognition difficult. Mapping letters to simplified recognition-optimised gestures facilitates improved recognition rates, but requires teaching and memorization of the specialized character set, prior to use. This learning requirement is impractical for "walk-up and use" systems. Virtual keyboards provide a suitable compromise between reliable recognition and minimal user learning. Text can be input using a virtual keyboard by navigating to characters using a small set of consistent navigational gestures. This makes virtual keyboards a good solution to the common problem of the available

gesture set having fewer unique members than contained in the required character set. To enable high performance with untrained use of virtual keyboards, there should be an intuitive mapping of gestures to navigation commands.

The primary disadvantage of virtual keyboards is the requirement for a visual interface. This potentially reduces the mobility of the device due to the required display's footprint. The user may also require their visual attention to be focused elsewhere. As with marking menus, once the user is familiar with the gesture set and relative locations of characters, less visual attention may be required [7, 21].

**Character Set.** The context of use determines the required character set. One-to-one mappings of characters to keys on a virtual keyboard can result in a larger interface footprint than desired. A large keyboard can also lead to longer acquisition times as the navigation distance increases. One way to resolve this problem is to organize characters into sub-groupings or hierarchies, resulting in a many-to-one mapping of characters to virtual keys. Characters are subsequently selected through a divide and conquer technique. Predictive text engines (e.g. T9 [9]) may be used to disambiguate characters. However, for applications where the occurrence of non-dictionary words is high (e.g. entering names on a game console) manual disambiguation (through additional gesture input) is still required.

**Layout Optimization.** Character arrangement on a virtual keyboard can be optimized according to the context of use. Familiar layouts such as alphabetical or QWERTY are best suited for walk up and use situations. These layouts require less visual scan time, providing more efficient operation. Optimizing the layout based on the frequency of letter selection may improve the performance of expert users as frequently used characters can be positioned such that travelling distance is minimised and/or less error-prone gestures are required for selection.

**Feedback.** Closed feedback loops can result in more accurate text entry as users have the opportunity to learn from and correct mistakes. Open loop feedback only allows identification of errors once they have occurred. Visual feedback is the typical channel for this communication; however, it may be enhanced or replaced by auditory and haptic feedback depending on the context of use. Mouse cursor style immediate, continuous visual feedback should be avoided due to the technical limitations of accelerometers (velocity variations can lead to non uniform mapping of tilt to cursor motion). When remote visuals are provided, designers should be aware of the potential problems of users exhibiting a split focus of attention and a decreased ability to correlate their motions with the visual representation.

## LIMITATION ANALYSIS OF WRIST BASED GESTURES

GesText systems are based purely on gestures made with the wrist. To aid the design of such systems we performed two pilot studies to investigate the physical limitations of human input when making such gestures. We focus on two

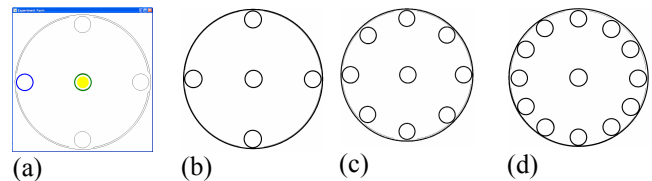
aspects of these gestures: 1) *Angle of selection*: can users differentiate different angles around a two-dimensional plane? 2) *Depth of selection*: can users make selections at different depths along a single axis?

### Pilot Study One: Angular Selections

The aim was to determine whether users can quickly and accurately select positions around a two-dimensional plane and the minimum differentiable angle between these points. Users were cued to make gestures to a particular position using three different layouts (see Figures 2b–2d).

#### Participants and Apparatus

Six volunteers (two female) participated in the experiment, two of whom were left-handed. Their mean age was 24.2 years (s.d. 1.7 years, range 22–26). Participants used a standard Wiimote to perform the gestures as it is a generic tilt-based device. Only the accelerometer was used for sensing the user's actions. Three of the participants had previous experience using accelerometer-based controllers (on video games or phones). The study lasted around 30 minutes.



**Figure 2: (a) Experimental interface; Target layouts, from b-d, with 90°, 45° and 30° separation.**

#### Experimental Interface and Task

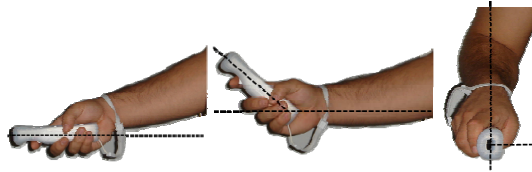
The experimental interface (see Figure 2a) had a size of 640×640 pixels and was displayed on a 32 inches wide-screen monitor (resolution 1280×768 px). Participants stood 2m from the screen. Each trial begins with the target position highlighted with a blue circle. A timer counts down from three to one over a period of one second. During this time, the participant is required to hold the Wiimote still, at the center or neutral position. Once the countdown completes, the user performs the required gesture (as described below). The participant is automatically moved to the next trial after completing the gesture (repetition was not required for incorrect gestures). The experimental interface cued all tasks and logged all task completion times.

To complete vertical gestures, the user tilted their wrist along the y-axis (Figure 3, center), dwelled in that position for at least 20ms (determined by informal trials) and returned to the center. Horizontal gestures were made by rotating the wrist around the z-axis (Figure 3, right) and completing a similar dwell and return routine. Gestures that did not fall on the horizontal or vertical axes were made using combinations of these base gestures. The required gestures were indicated on-screen using a circular layout with the target highlighted using a blue circle (see Figure 2a).



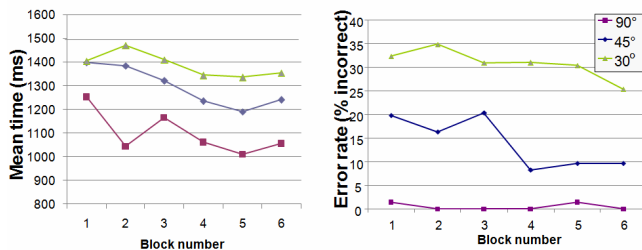
### Design and Procedure

Task completion time was analysed using a 3×6 within-factor repeated measures analysis of variance (ANOVA), with the two factors: *separation angle* (90°, 45°, 30°) and *block number* (1–6). The number of errors were also recorded and analysed separately.



**Figure 3: Wrist movements (left) neutral position - holding wrist in a horizontal orientation to target the center point; (center) tilting along the y-axis to target characters above or below the center point; (right) rotating the wrist around the z-axis to target characters to the right or left of the center point.**

The experiment was divided into three parts, one for each of the three layouts described above. For each part, the participants first completed a training session, consisting of a freeform practice session (where users could complete any gesture) and then a cued practice session, where the user completed one block of gestures, such as would be encountered during the timed sessions. All data from the practice sessions was discarded. Then, each user completed six blocks with a particular layout. Each block required the user to perform all gestures for that layout, three times, in a random order. Once the user had finished all blocks for one layout, a NASA-TLX form was completed for that layout and the procedure repeated for the next layout. The order the layouts were provided to the users was also randomised.



**Figure 4: (left) Mean target acquisition times, for pilot study one. (right) Error rates. Error bars omitted for clarity.**

### Results and Discussion

The results from the first pilot study are summarised in Figure 4, left. A total of 2604 selections were made during the experiment, 115 of these were discarded as they took longer than +3 s.d. from the mean time. There was a significant effect for *separation angle*, ( $F_{2,10}=4.9, p < 0.05$ ) with best performance with targets in the 90° separation layout (mean=1.1sec, s.d. 0.3sec). This was followed by the 45° separation layout (mean=1.3sec, s.d. 0.3sec) and then 30° (mean=1.4sec s.d. 0.3sec). As expected, there was also a significant difference between blocks, with the first block taking a mean time of 1.4sec (s.d. 0.3sec) and the last block at mean time of 1.2sec (s.d. 0.3sec) ( $F_{5,25}=9.6, p < 0.01$ ). The block/layout interaction ( $p < 0.05$ ) is caused by the differing rates of learning through the blocks of each layout.

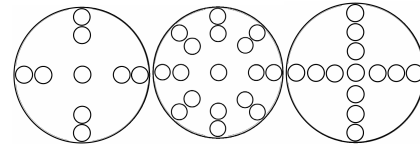
Error rates for the three layouts are summarized in Figure 4, right. The target acquisition time and the error rates increased as the separation angle decreased (and the number of items increased). Participants had little or no errors at 90° separation angle (0.46% errors), but significantly more at 45° (14% errors) and 30° layouts (31% errors),  $F_{2,10} = 32.5, p < 0.01$ . There was also a significant difference for *block number*, with the first block having a mean error rate of 18% and the last block 12% ( $F_{5,25}=3.8, p < 0.05$ ). Participants subjectively rated the 90° layout the least effort (across all NASA-TLX categories) and best performing.

### Pilot Study Two: Depth Selections

The goal of the second pilot was to determine whether users could quickly and accurately select targets at a certain depth. It used the same apparatus and procedure as pilot study one, with the participants, gestures and experimental design differing as described below.

### Experimental Design and Participants

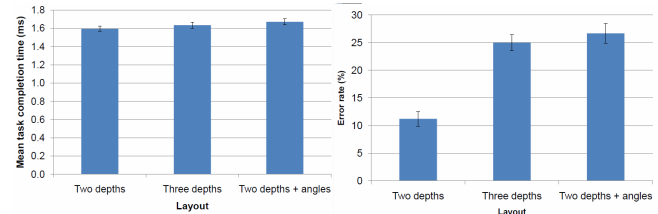
The task-time dependent measure was analysed using a 3×6 repeated measures analysis of variance (ANOVA), with the two factors: *depth* (2, 2+angles,3) and *block number* (1–6). Six volunteers (two female) participated in the experiment. Their mean age was 29 years (s.d. 6.8 years, range 22–40) and all were right-handed.



**Figure 5: Layouts two deep, two deep+angles and three deep.**

### Depth Gestures

Depth gestures were completed in a similar manner to that described earlier. Figure 5 shows the three layouts used in this experiment. Different depths were selected by extending or flexing the wrist along the y-axis with the Wiimote.



**Figure 6: (left) Task completion times for pilot study two; (right) Error rates.**

### Results

There was no significant main effect for the factors layout or block number on task completion times. The two-depth layouts and the three depth layout had a mean time of 1.6sec (both with s.d. of 0.2sec) and the two depths+angles had a mean time of 1.7sec. These are summarized in Figure 6, left. There was a significant difference in error rates between the layouts. Two depths had the lowest errors, with a mean of 11%, with three depths and two depths+angles

having error rates of 25% and 27% respectively ( $p < 0.01$ ,  $F_{2,10}=9.7$ ). Errors did not vary significantly over blocks.

#### *Summary of Pilot Study Two*

While there was no significant difference between target selection times between these three layouts, the error rates show that users struggle with greater than two depths and also with the combination of depths and angles.

#### **Discussion of Pilot Studies**

The design implications for GesText systems from the observations made during the pilot studies are now discussed.

##### *Increasing the Number of Available Selection Points*

Increasing the number of gestures by including depths and diagonals significantly increased the number of errors. Pilot study one found that single horizontal and vertical gestures were considerably quicker and less error prone. However, an text-entry system must provide selection possibilities for all alphabetic characters (26 for English), a space and back-space character (and possibly punctuation and numeric characters). To do this, a mechanism for providing these extra selection points must be chosen. We considered two possibilities: adding depth and adding diagonals.

When participants were required to select between multiple depths along an axis, outside points (those furthest from the center with effectively infinite boundaries) generally fared better than the inner and central points. One possible solution is to increase the selection area for the internal points. This would provide a similar effect to that of increasing the target size in a Fitts' task.

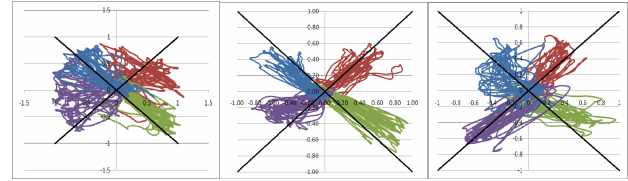
Examining the selection path data for diagonal gestures reveals the extent of the problem. Figure 7 shows the gesture paths for three different participants. The black diagonals show the idealised gesture directions. For the right-handed participants (left and center figures), the bottom right and top left positions more closely adhere to the diagonal axes than the opposite two diagonals. Interestingly for left-handed participants this result is mirrored (the bottom-left and top-right are easier).

Difficulties in selecting diagonal targets are potentially due to the motion that is required along multiple axes, simultaneously. The radial nature of the layout means that the selection area of vertical and horizontal gestures is reduced by the inclusion of diagonals. Inaccurate axial movements can subsequently be interpreted as diagonal gestures, resulting in an increased error rate. Where these selection errors detrimentally impact the overall text entry rates it may prove beneficial to transform these gestures into successive motions, forming compound gestures. Compound gestures are discussed further in the design of our GesText system.

##### *Gesture Selection and Delimiting*

Using dwell time to select targets has the disadvantage of requiring the user to accurately position themselves over a node for a period of time. Participants frequently reported

that if they overshoot the desired target, they did not have enough time to correct their mistake before the erroneous target was selected. Further increasing the dwell time required before selection would further increase the per-character selection time. Cumulatively these several-hundred millisecond pauses could noticeably impact the speed obtainable by a gestural text entry system.



**Figure 7: Gesture path plots. Solid black diagonals represent idealized positions. (left) Right-handed participant with a high error rate, yet the bottom right is still clearly easier. (centre) Right-handed participant with a low error rate. The bottom right and top left are easier. (right) Left-handed participant with medium error rate. The bottom-left and top-right are easier (mirrored from the right-handed participants).**

We also frequently observed participants having difficulty returning to the central position after performing a gesture. This may partially be due to the split-focus required: users were concentrating on the screen which displayed instructions, meaning they were unable to look down at their hand to accurately locate the horizontal position. The feedback provided for locating the central position therefore impacts on the ability of users to accurately perform the required gestures. A viable solution to this problem is to increase the size of the central area.

#### **DESIGN OF MID-AIR TEXT ENTRY SYSTEMS**

From the pilot studies, we learnt that: 1) increasing the number of gestures by including depths and diagonals significantly increased the number of errors, 2) dwell based selection methods result in a lack of error correction ability and impact selection time, 3) returning to and remaining at the central point between selections is sometimes difficult, 4) the accuracy and drift of diagonal gestures showed large differences between users. Mirroring of difficult gestures is also visible between right- and left-handed people. Using these findings as input, and considering the design factors outlined earlier, we designed two contending GesText systems. Both designs incorporate the three fundamental lessons derived from the pilot studies, and are detailed below.

#### **System enhancements**

##### *Multiple centers*

Multi-center layouts offer an attractive solution to the higher error rates and acquisition times found in the pilot studies for depth-based and angular gestures. Users can still employ the quick and (almost) error-free vertical and horizontal position selections, and can instead employ multiple gestures to select a character. The primary disadvantage of this technique is the increased selection time (two or more gestures required instead of one). However, this is an ac-

ceptable compromise when compared to the increased selection time and frustration experienced when including highly error-prone gestures.

#### *Compound gestures*

Restricting gestures to axial movements around multiple centers is an accurate yet inefficient way of meeting the demand for gestures. Including diagonal selections significantly reduces the number of centers required. However, the pilot studies showed that the selection of diagonal gestures was more error prone than the horizontal and vertical axes. To resolve this issue we employ compound gestures in the text entry systems as a method of selecting diagonal items. A compound gesture breaks a diagonal movement into its constituent components: a vertical gesture and a horizontal gesture. In-house testing showed this method to be more accurate than the single diagonal gesture.

#### *Selection Mechanism*

We propose a rapid-return-to-center gesture as an alternative to dwelling for selection. This resolves the lack of error correction and increased selection time, present in the latter. All of the investigated gestures require the user to begin at a central point, gesture to the correct location and return to the central point. This action of returning to the center can therefore be used as a gesture delimiter; once a user has highlighted their desired target. This has the triple advantage of eliminating dwell time delays, making it easier for users to correct from overshoot mistakes and ensuring that the user automatically returns to center; enabling the rapid performance of successive gestures.

#### **Core similarities of the two proposed GesText systems**

Both of the GesText systems incorporate a virtual keyboard from which characters are selected using wrist-based navigational gestures. The character set comprises the 26 English alphabet characters, along with space and delete. The layout of the two systems vary due to the different techniques used to accommodate the 28 characters. Character disambiguation is achieved manually using additional gestures rather than through predictive text engines, due to the intended context of use; novice users entering non-dictionary words (such as names on a games console). The alphabetical arrangement of characters for both systems is also implemented as an optimization for novice users.

#### **Layout 1: Matrix**

The matrix-based layout primarily uses single depth vertical and horizontal gestures and single depth compound diagonal gestures for character selection, as shown in Figure 1, left. The complete character set is provided by using multiple gestures for each character.

Character selection begins in the central square. For alphabetic characters, the user first gestures towards the subset square that contains the required letter (this is done with a gesture into the required square and a return to center). The cursor then moves into the required subset. Once inside, the

process is repeated for selecting the required letter. Any blank square can be used as an 'escape' to return to the center. On character selection, the cursor returns to the matrix's center. Diagonal gestures are achieved using compound gestures, as described earlier. The selection of backspace and space characters requires only a single gesture.

#### **Layout 2: Tri-Center**

The tri-center layout (Figure 1, right) uses three centers for character layout and selection. It uses single depth compound diagonal gestures and a combination of single and double depth horizontal and vertical gestures for character selection.

Character selection begins in the middle center-point (indicated by a star symbol). From this position, the user can select any of the nine characters immediately surrounding the center-point, or the vertical double-depth functions of space and backspace. To select characters centered around the left-hand center, the user makes a double-depth motion to the left, moving the center-point. The user can now select characters as per the middle center-point. This layout supports center selection correction (in case the incorrect center is selected), but does not support correction once a gesture towards a character has begun.

#### **EXPERIMENT ONE: WALK-UP USE OF ACCELEROMETER-BASED TEXT ENTRY SYSTEMS**

This experiment investigates the performance of the two accelerometer-based text-entry systems described in the previous section. Participants were asked to enter a series of short sentences using each of the two interfaces.

#### **Participants and Apparatus**

Fifteen volunteers (five female) participated in the experiment, three of whom were left-handed. Their mean age was 24.1 years (s.d. 2.4 years, range 19–28). As per the pilot studies, participants used a Wiimote for gesture entry (and again, only the accelerometer was used for sensing the user's actions). Only one participant indicated that they had no prior experience using accelerometer-based controllers. Participants entered text using the two gesture-based interfaces described in the previous section. The matrix-based interface was  $517 \times 515$  pixels, while the tri-center interface was  $710 \times 325$  pixels. They were displayed on the 32 inches monitor 2m in front of the participants. The experiment lasted approximately 45 minutes.

#### **Tasks**

Each task required the user to enter the provided sentence with the given interface. The participants were given 10 seconds to memorise the sentence before the interface was revealed, with the sentence the participant was required to enter displayed above the interface. The entered characters appeared immediately below the required sentence. Incorrect entries caused a system 'beep' and the entered character to be shown in red. Participants were required to correct their mistakes before the task could end.



Two sets of 17 sentences were randomly selected from MacKenzie and Soukoreff's publically available corpus (<http://www.yorku.ca/mack/chi03b.html>). The first set was always used first, regardless of the interface. Sentences were presented in a random order to the users. The input time was measured from the beginning of the first gesture to completion of the final character.

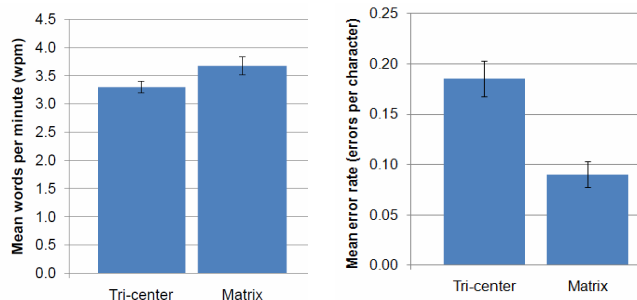
### Experimental Design and Procedure

Typing speed is analysed with a repeated measures ANOVA with interface type as the factor (matrix and tri-center). The number of errors per sentence was also recorded and analysed separately.

The experiment was divided into two parts, one for each of the two layouts described earlier. For each interface, the participants first completed a training session. The training consisted of a freeform session where the user could select any character they wished (lasting five minutes) and then two cued sentences, similar to that which they would complete during the recorded tasks. All data from the practice sessions was discarded. Following the practice sessions, each participant 'typed' fifteen sentences using their first interface. A subjective feedback form, including a NASA-TLX evaluation was completed at this point. The procedure was then repeated for the remaining interface. The order the interfaces were provided to the users was counter-balanced, with eight using the matrix layout first and seven using the tri-center layout first.

### Results

The results from experiment one are summarised in Figure 8 (left). A total of 225 sentences were 'typed' during the experiment. There was a significant difference for factor *interface*, with the matrix interface having a higher typing speed with 3.7wpm (s.d. 0.6wpm) and the tri-center interface having 3.3wpm (s.d. 0.4wpm),  $p < 0.05$ ,  $F_{1,14} = 10.1$ .

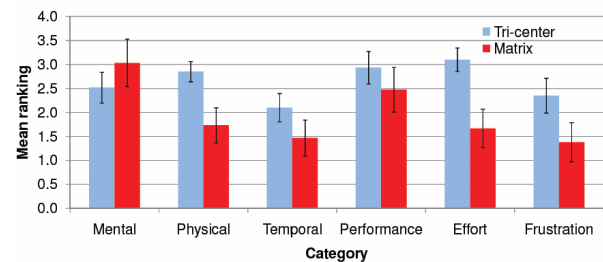


**Figure 8: (left) Mean character entry rate for the two interfaces; (right) Mean error rates for each interface**

The error rates for the two interfaces are summarized in Figure 8 (right). There was a significant difference between the error rates for each layout, with the matrix-based layout having the fewest errors (mean 0.09 errors per character, s.d. 0.05) and the tri-center the greatest (0.19, s.d. 0.06),  $p < 0.01$ ,  $F_{1,14} = 42.2$ .

### Subjective Analysis

After completing the tasks with each interface, the participants completed NASA-TLX evaluation forms. At the conclusion of the experiment overall preference rankings were also gathered. The results of the NASA-TLX surveys are shown in Figure 9. In all categories, except mental demand, the matrix-based interface out-performed the tri-center interface. Preferences for the two interfaces were significantly different, with the matrix-based interface preferred by 11 participants, the tri-center interface preferred by one participant and three could not decide: Wilcoxon  $z = 2.35$ ,  $p < .01$ .



**Figure 9: Experiment one NASA-TLX results**

### Summary of Experiment One

The results of experiment one showed that the matrix-based layout was significantly faster and significantly less error-prone than the tri-center layout. Subjective results indicated that participants also preferred the matrix-based layout.

This study has shown for novice, 'walk up and use' situations, the matrix-based interface provides the best performance, with the least number of errors. To explore whether this result holds over longer periods of time, we conducted a study with repeated use over several days.

### EXPERIMENT TWO: LONGTERM LEARNING IN TEXT ENTRY SYSTEMS

This experiment investigates how the performance (efficiency and error rates) of the two text-entry systems is affected by experience. Participants were asked to use the two GesText systems daily for a period of four consecutive days. The apparatus and experimental design are the same as for experiment one with the participants and procedure varying as described below.

#### Participants

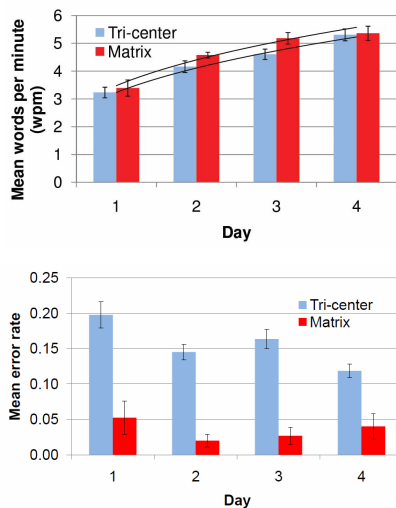
Five volunteers (two female) participated in the experiment, one of whom was left-handed. Their mean age was 22 years (s.d. 1.7 years, range 19–23). All participants had experience using accelerometer-based controllers (for video games or on phones). The experiment lasted approximately 45 minutes.

#### Experimental Design and Procedure

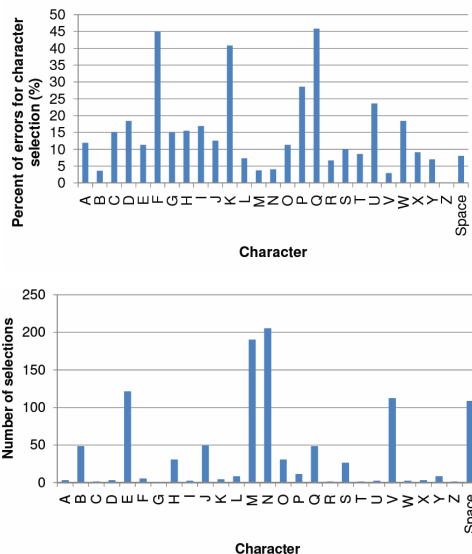
The typing speed dependent measure was analysed using a  $2 \times 4$  factor repeated measures analysis of variance (ANOVA), with the factors: *interface* (matrix and tri-

center) and *day* (1–4). The number of errors per sentence were also recorded and analysed separately.

This experiment followed the same procedure as experiment one, with the exception of the collection of subjective data: participants were only asked to complete this information on the first and last days.



**Figure 10: (top) Mean words per minute for experiment two (bottom) Error rates for experiment two.**



**Figure 11: Aggregated error data over the four days (top) Tri-center percentage error rates per character (bottom) Tri-center incidences of wrongly selected characters**

## Results

A summary of the rate of word entry for the two interfaces, over the four days, is shown in Figure 10, top. There was a significant difference for the factor *days*, with the first day having a mean of 3.3wpm (s.d. 0.5wpm) and the fourth day 5.3wpm (s.d. 0.5wpm),  $p < 0.01$ ,  $F_{3,12}=55.9$ . As can be seen in the figure, there is a power law fit to the learning rates of both interfaces ( $R^2=0.99$  for tri-center and  $R^2=0.97$  for ma-

trix), with the greatest learning occurring between days one and two. No other factors showed significant differences.

The error rates for the two interfaces are summarized in Figure 11. There was a significant difference for the factor *day* ( $p < 0.01$ ,  $F_{3,12} = 6.4$ ). The tri-center layout exhibited a significantly higher error rate (0.15, s.d. 0.04) than the matrix-based layout (0.07, s.d. 0.04),  $p < 0.01$ ,  $F_{1,4}=89.7$ . For both interfaces, the error rate was the highest on the first day and was variable for the remaining days. Significant variation of the percentage error rates per-character and the incident rate of wrongly selected characters is apparent for the tri-center layout, as shown in Figure 11.

## Subjective Analysis

All five participants indicated that they preferred the matrix-based interface. The NASA-TLX results showed that participants also thought that their performance increased over the four days. On the final day of experiments, participants expressed that the tri-center layout “improves significantly with use” explaining that “the knowledge of how to use the system is intrinsic, but the feel for it takes time”.

## Summary of Experiment Two

This experiment has shown that the performance of an accelerometer only, gesture-based text-entry system does improve with regular use. Users can increase their entry rate by up to 65% from their first day, after four days of use.

## DISCUSSION

The matrix-based layout is more suitable for novice users than the tri-center layout, as the error rate of the tri-center layout remains comparably higher. An investigation into the causes of the various observed errors and difficulties exhibited by participants highlighted a few specific areas, where if improvements were made, considerable performance benefits could be expected, particularly for the tri-center layout. These are discussed in this section.

### Increasing the Number of Available Selection Points

Due to the limited size of the available gesture set (8 unique gestures) it was necessary to use a combination of compound gestures, multiple depth gestures and multiple centers to increase the number of available selection points.

**Compound Gestures.** The compound gesture technique used in both layouts was less error prone than the one stage diagonal gestures employed in the pilot studies. However the majority of participants requested the ability to choose the order to perform the sub-gestures, e.g. move up or down after rolling left or right as well as vice versa.

**Multiple Depth Gestures.** Examination of Figure 11 reveals that in the tri-center layout, where gestures incorporated a multi-depth selection, errors rates were significantly higher. An example of this is between characters E and space and also D and F, due to the first step of the compound gesture. Participants found double depth “irritating because you have to stop and be careful ... therefore it is frustrating as it interrupts the rhythm of inputting text”. The central charac-

ters only require one gesture and should, in theory, be much quicker to select. If the problem of having to gesture slowly to avoid overshooting was eliminated, a higher WPM rate could be achieved. For experienced users, breaking the alphabetical layout to place frequently used characters around the main central point could then subsequently further improve the performance of the tri-center layout.

Following the difficulties finding and maintaining the center position observed in the pilot studies, the centers were enlarged to reduce random motion errors. This aim was successfully achieved; however this appears to have exacerbated the multiple depth problem. The discrete visual feedback employed in both layouts (whereby the cursor jumps between potential targets) coupled with the size of the center, meant that a noticeable displacement gesture could be performed whilst the cursor was still at the center.

**Multiple Centers Involving Cursor Warping.** Another common problem on the tri-center layout was “overcompensating the return roll after selecting a letter from the outer side blocks” resulting in selecting an extra M or N. This problem can clearly be seen in the incident rates of wrongly selected characters in Figure 11, bottom. The temporary non linear mapping of gestures that occurs when the cursor warps back to the main center following the selection of characters from the outer edges could have caused confusion resulting in these errors. Disregarding a few sensor readings immediately preceding such gestures may be a viable solution to this problem.

## CONCLUSION

This paper has provided a description of the three categories of factors that influence the design of accelerometer –based text-entry systems; human, accelerometer and interface considerations. The limits of human ability to perform wrist-based gestures with accelerometers were evaluated, leading to the establishment of GesText; a class of novel text-entry designs using a set of simple tilt based gestures.

We evaluated the performance of two GesText systems with respect to user efficiency and subjective satisfaction, demonstrating that accelerometer only text input is viable. Overall, the matrix layout was found to be most suitable for novice users, demonstrating a significantly higher WPM rate, user preference and a lower error rate. The performance of the tri-center interface improves with experience, as users learn the scale of gesture required to select double depth characters. This style of interface is best suited to situations where people are willing to put in the effort required to become proficient. Both systems are suitable for inputting small amounts of text (e.g. as on games consoles).

Future work in this area will focus on evaluating these gestural interfaces for different input devices (e.g. joysticks or gyroscope-based devices) and exploring other methods for distinguishing between multiple-depth gestures (e.g. using additional feedback channels).

## REFERENCES

1. Crossan, A. Williamson, J., Brewster, S. and Murray-Smith, R. (2008) Wrist rotation for interaction in mobile contexts. *Proc. MobileHCI '08*, 435-438.
2. Eslambolchilar, P. and Murray-Smith, R. (2004) Tilt-based automatic zooming and scaling in mobile devices - a state-space implementation. *Proc. MobileHCI '04*, 120-131.
3. Grandjean, E. (1969) Fitting the task to the man: an ergonomic approach. Taylor and Francis, p. 372.
4. Hinckley, K., Baudisch, P., Ramos, G., and Guimbretiere, F. (2005) Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. *Proc. CHI '05*, 451-460.
5. Hinckley, K., Pierce, J., Sinclair, M. and Horvitz, E. (2000) Sensing techniques for mobile interaction. *Proc. UIST '00*, 91-100.
6. Igarashi, T. and Hinckley, K. (2000) Speed-dependent automatic zooming for browsing large documents. *Proc. UIST '00*, 139-148.
7. Kurtenbach, G., Sellen, A. & Buxton, W. (1993). An empirical evaluation of some articulatory and cognitive aspects of "marking menus". *Journal of Human Computer Interaction*, 8(1): 1-23.
8. Lee, H., Khandelwal, M. and Mazalek, A. (2007) Tilting table: a movable screen. *Proc. TEI '07*, 93-96.
9. MacKenzie, I. S., and Tanaka-Ishii, K. (Eds.) Text entry systems: Mobility, accessibility, universality, pp. 332. San Francisco.
10. Mobitrends, [www.mobitrends.com/category/gadgets/page/3/](http://www.mobitrends.com/category/gadgets/page/3/)
11. NASA (1995) NASA-STD-3000 Man-Systems Integration Standards. Rev B.
12. Oakley, I. and O'Modhrain, S. (2005). Tilt to scroll: Evaluating a motion based vibrotactile mobile interface, *Proc. WHC '05*, 40-49.
13. Oakley, I. and Park, J. (2007) A motion-based marking menu system. *CHI '07 Extended Abstracts*, 2597-2602.
14. Partridge, K., Chatterjee, S., Sazawal, V., Borriello, G. and Want, R. (2002) TiltType: Accelerometer-supported text entry for very small devices. *Proc. UIST '02*, 201-204.
15. Rahman, M., Gustafson, S., Irani, P., and Subramanian, S. (2009) Tilt techniques: investigating the dexterity of wrist-based input. *Proc. CHI '09*, 1943-1952.
16. Rekimoto, J. (1996) Tilting operations for small screen interfaces. *Proc. UIST '96*, 167-168.
17. Sazawal, V., Want, R., and Borriello, G. (2002). The Unigesture Approach. *Proc. MobileHCI '02*, 256-270.
18. Shoemaker, G., Findlater, L., Dawson, J. Q., and Booth, K. S. (2009) Mid-air text input techniques for very large wall displays. *Proc. Graphics interface '09*, 231-238.
19. Weberg, L., Brange, T. and Hansson, Å. W. (2001) A piece of butter on the PDA display. *Proc. CHI '01 Extended Abstracts*, 435-436.
20. Wigdor, D. and Balakrishnan, R. (2003) TiltText: Using tilt for text input to mobile phones. *Proc. UIST '03*.
21. Zhao, S. and Balakrishnan, R. (2004). Simple vs. Compound Mark Hierarchical Marking Menus. *Proc. UIST '04*, 33-42.